

**IN THE CLAIMS:**

The following listing of claims will replace all prior listings of claims in the application:

1. (Currently Amended): A method for network protocol filtering of a packet using an address resolution table that is cross-linked with a state table indexed with an address resolution table index (ART index), the packet having a Media Access Control (MAC) destination address, the method comprising:
  - determining a packet type for the packet;
  - obtaining packet information for the packet including the MAC destination address;
  - determining that the MAC destination address is included in an entry in the address resolution table;
  - obtaining the ART index associated with the MAC destination address from the entry in ~~based on~~ the address resolution table, wherein the ART index ~~obtained~~ is an index into the state table for locating an entry in the state table; and
  - storing the ~~obtained~~ ART index and the packet information in a data structure associated with the state table.
2. (Currently Amended): The method, according to claim 1, further comprising:
  - determining whether the packet is for a new connection; and
  - responsive to the packet not being for the new connection, ~~[[the]]~~ determining whether the packet information is in the address resolution table.
3. (Original): The method, according to claim 2, wherein the packet type is a Transmission Control Protocol type.
4. (Original): The method, according to claim 1, wherein the packet type is a User Datagram Protocol type.
5. (Original): The method, according to claim 1, wherein the packet information is a five-tuple including source and destination addresses, source and destination ports, and a packet type identifier.

6. (Original): The method, according to claim 1, wherein the packet type is a Generic Routing Encapsulation type.

7. (Original): The method, according to claim 6, wherein the packet information is a five-tuple including source and destination addresses, an apportioned Generic Routing Encapsulation identifier, and a packet type identifier.

8. (Original): The method, according to claim 1, wherein the packet type is an Internet Protocol Security type.

9. (Original): The method, according to claim 8, wherein the packet information is a five-tuple including source and destination addresses, an apportioned security parameter string, and a packet type identifier.

10. – 13. (Cancelled)

14. (Currently Amended): A method for inbound network address translation packet filtering using an address resolution table that is cross-linked with a state table indexed with an address resolution table index (ART index), the packet having a Media Access Control (MAC) destination address, the method comprising:

- obtaining a packet;

- determining whether type of the packet is one of a Transmission Control Protocol, a User Datagram Protocol, a Generic Routing Encapsulation, an Internet Protocol Security and an Internet Control Message Protocol type;

- if the type is the Transmission Control Protocol type, determining if the packet is an initial packet for a connection;

- if the type is the Transmission Control Protocol type and the packet is for an existing connection or if the type is one of the User Datagram Protocol type, the Generic Routing Encapsulation type and the Internet Protocol Security type,

- obtaining packet information from the packet including the MAC destination address;

- determining that the MAC destination address is included in the address resolution table;

obtaining the ART index associated with the MAC destination address from the entry in ~~based on~~ the address resolution table, wherein the ART index ~~obtained~~ is an index into the state table for locating an entry in the state table; and

storing the ~~obtained~~ ART index and the product information in the data structure associated with the state table.

15. – 18. (Cancelled)

19. (Previously Presented):        The method, according to claim 14, further comprising:

checking validity of layers of the packet;  
checking Internet Protocol options for the packet; and  
determining whether the packet is a fragment.

20. – 26. (Cancelled)

27. (Previously Presented):        The method, according to claim 14, wherein the data structure is for a plurality of canonical frame headers.

28. (Cancelled)

29. (Previously Presented):        The method, according to claim 14, wherein the state table is a connection table.

30. (Cancelled)

31. (Previously Presented):        A method for outbound packet filtering using an address resolution table that is cross-linked with a state table indexed with an address resolution table index (ART index), the packet having a Media Access Control (MAC) destination address, the method comprising:

obtaining a packet;  
determining whether an incoming interface for the packet is running network address translation;  
if the incoming interface is running the network address translation,

obtaining a first index from a data structure associated with the packet;  
and  
obtaining packet information in a first table using the first index;  
determining whether type of the packet is one of a Transmission Control Protocol, a User Datagram Protocol, a Generic Routing Encapsulation, an Internet Protocol Security and an Internet Control Message Protocol type;  
if the type is the Transmission Control Protocol type, determining if the packet is an initial packet for a connection;  
if the type is the Transmission Control Protocol type and the packet is for an existing connection or if the type is the User Datagram Protocol type,  
obtaining the packet information from the packet including the MAC destination address,  
determining that the MAC destination address is included in the address resolution table,  
obtaining the ART index associated with the MAC destination address from the entry in ~~based on~~ the address resolution table, wherein the ART index ~~obtained~~ is an index into the state table for locating an entry in the state table,  
and  
storing the ~~obtained~~ ART index and the packet information in a data structure associated with the state table;  
if the type is the Internet Control Message Protocol type, determining whether the Internet Control Message Protocol type is on a list of Internet Control Message Protocol types;  
if the type is not the Internet Control Message Protocol type,  
determining if the outgoing interface is running the network address translation;  
responsive to the outgoing interface running the network address translation,  
obtaining the second index from the data structure; and  
obtaining the packet information from the first table using the second index.

32. – 35. (Cancelled)

36. (Original): The method, according to claim 31, wherein the packet information is a five-tuple including source and destination addresses, source and destination ports, and a packet type identifier.

37. (Original): The method, according to claim 31, wherein the packet type is a Generic Routing Encapsulation type.

38. (Original): The method, according to claim 37, wherein the packet information is a five-tuple including source and destination addresses, an apportioned Generic Routing Encapsulation identifier, and a packet type identifier.

39. (Original): The method, according to claim 31, wherein the packet type is an Internet Protocol Security type.

40. (Original): The method, according to claim 39, wherein the packet information is a five-tuple including source and destination addresses, an apportioned security parameter string, and a packet type identifier.

41. (Original): The method, according to claim 31, further comprising:  
checking validity of layers of the packet;  
checking Internet Protocol options for the packet; and  
determining whether the packet is a fragment.

42. – 51. (Cancelled)

52. (Previously Presented): The method, according to claim 31, wherein the data structure is for a plurality of canonical frame headers.

53. – 66. (Cancelled)